

ERASURE-AND-SINGLE-ERROR CORRECTION DECODER FOR LINEAR BLOCK CODES

BACKGROUND

Field

[1001] The present invention relates generally to data communication, and more specifically to techniques to efficiently perform erasure-and-single-error correction decoding for linear block codes.

Background

[1002] With the advent of digital communication and the need to transmit large amounts of data through an impaired and band-limited channel, the need for coding of data to facilitate correct reception is of great importance. A data transmission is typically degraded by impairments in a communication channel, such as thermal noise, interference, spurious signals with the transmission bandwidth, and so on. The received data is thus typically a distorted version of the transmitted data.

[1003] Coding may be used to allow a receiver to detect and/or correct for errors in the received data. Various error-correction codes are available and may be categorized into several classes such as block codes and convolutional codes. Convolutional codes provide good error-correction capability but typically output correlated bursts of errors. Block codes have built-in burst error handling capability when combined with a proper level of interleaving. For example, a Reed-Solomon code can handle any burst of errors within a symbol, which may be defined as comprising a particular number of bits.

[1004] In theory, a block code is able to correct for a particular number of erasures and/or a particular number of errors, with the exact number for each being determined by the distance of the code. An erasure may be indicated for a symbol that is known *a priori* to be potentially bad, and an error is a symbol that is received in error but is not known as such *a priori*. Erasures are typically known or may be determined by the receiver, and may be accounted for accordingly in the decoding process. Errors are undetected symbol errors, which may be symbols that are erroneously detected as having been received correctly when in fact they were not.

[1005] Conventional erasure-and-error correction block decoders (such as the Berlekamp-Massey or Euclidean decoder) are complex and typically require implementation in dedicated hardware. These block decoders are typically too computationally intensive for software-based implementation executed on a microprocessor. Hardware based decoders can exploit parallelism in the decoding algorithms and use a pipelined datapath, both of which are not possible on a traditional microprocessor. More efficient block decoder algorithms are available that may be more suitable for software-based implementation. However, these decoder algorithms typically have limited capabilities and may be able to correct for erasures but not errors.

[1006] There is therefore a need in the art for an efficient erasure-and-error correction decoder for linear block codes, and which may be suitable for software-based implementation.

SUMMARY

[1007] Aspects of the invention provide techniques for efficiently performing erasure-and-single-error correction block decoding on a received block of symbols previously coded column-wise with an (N, K) linear block code and row-wise with an error detection code. If the error detection code (e.g., a CRC code) on the rows has relatively good error detection properties, then the probability of more than one undetected error row in a received data block is very low. Thus, single error correction is typically sufficient in most applications.

[1008] Initially, each row of the received block is marked as either an erased row or an un-erased row until at least $(K+1)$ un-erased rows are found. Typically all N rows of the received block are marked, but this is not absolutely necessary. The rows may be marked based on cyclic redundancy check (CRC) bits included in each row, or by some other means. Erasure-only or erasure-and-error correction block decoding may then be performed on the received block depending on the number of erased rows, i.e., as supported by the distance of the (N, K) block code.

[1009] To perform erasure-and-single-error correction block decoding on the received block, a codeword corresponding to a column of the received block containing an undetected symbol error is initially identified. The undetected symbol error may be one that is not detected by the CRC check previously performed on each row of the received block. This codeword may be identified by (1) deriving an estimate of an un-

erased systematic row of the received block, (2) comparing the un-erased systematic row against its estimate, and (3) identifying an unmatched symbol between the un-erased systematic row and its estimate. The codeword then corresponds to the column containing the unmatched symbol.

[1010] The location of the (previously undetected) symbol error in the codeword is then determined based on a particular block decoding scheme (e.g., one known in the art and corresponding to the selected linear block code). The row of the received block containing the located symbol error is then marked as an erased row. Block decoding may then be performed in the normal manner for the received block with the marked erased row containing the symbol error. In this way, an erasure-only correction block decoding technique may be used to achieve erasure-and-single-error correction block decoding on the received block. The erasure-and-single-error correction block decoding technique described herein is computationally efficient (relative to conventional techniques) and may be implemented in hardware, software, or a combination thereof.

[1011] Various aspects and embodiments of the invention are described in further detail below. The invention further provides methods, computer program products, decoders, receiver units, and other apparatuses and elements that implement various aspects, embodiments, and features of the invention, as described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[1012] The features, nature, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

[1013] FIG. 1 is a block diagram of a transmitter unit and a receiver unit capable of implementing various aspects and embodiments of the invention;

[1014] FIG. 2 is a diagram that graphically illustrates the various steps for performing erasure-only correction block decoding for a linear block code;

[1015] FIGS. 3A and B graphically illustrate the various steps for performing erasure-and-single-error correction block decoding for a linear block code;

[1016] FIG. 4 is a simplified flow diagram of an embodiment of a process for performing erasure-and-single-error correction block decoding for a linear block code; and

[1017] FIGS. 5A and B show a detailed flow diagram of an embodiment of a process for performing erasure-and-single-error correction block decoding for an (N, K) linear block code.

DETAILED DESCRIPTION

[1018] The inventive block decoding techniques described herein may be used for various coding schemes. For clarity, these techniques are described for a specific two-dimensional product-coding scheme comprised of a cyclic redundancy check (CRC) row code and a linear block column code. An example communication system wherein the inventive block decoding techniques may be used is described below.

[1019] FIG. 1 is a block diagram of a transmitter unit 110 and a receiver unit 150 capable of implementing various aspects and embodiments of the invention. At transmitter unit 110, a data source 112 provides data (e.g., in frames of a particular length) to an outer encoder 120 that includes a block encoder 122 and a CRC encoder 124. Block encoder 122 receives and codes each block of a particular number of data frames to provide a corresponding block-coded data block. In an embodiment, the bits in a data block are first grouped to form symbols (with each symbol comprising N_B bits), and each column of K symbols in the data block is coded with a particular (N, K) linear block code to provide a corresponding codeword of N symbols. A block-coded data block thus comprises N rows of symbols. For a systematic block code, the first K rows are the data rows and the remaining $(N-K)$ rows are the parity rows.

[1020] For each of the N frames in the block-coded data block, CRC encoder 124 generates a set of CRC bits based on the data bits in the frame and appends the CRC bits to the end of the frame. The CRC bits included in each frame are used for error detection of the frame at the receiver unit. CRC encoder 124 provides coded blocks, each of which is CRC-coded in one (horizontal) direction and block-coded in the other (vertical) direction.

[1021] Block encoder 122 may implement any linear block code such as a Reed-Solomon code (which is commonly used for data transmission), a Hamming code, a BCH (Bose, Chaudhuri, and Hocquenghem) code, or some other code. The inventive

block decoding techniques described herein may be used for any linear block code and may advantageously be used for systematic block codes.

[1022] In the embodiment shown in FIG. 1, the coded blocks are provided to an inner encoder 130 that includes an interleaver 132 and a convolutional encoder 134. Interleaver 132 shuffles (i.e., reorders) the bits in each coded block and provides interleaved bits to convolutional encoder 134, which then codes the bits in accordance with a particular convolutional code. The interleaving provides time diversity and disperses errors that may occur in bursts from a convolutional decoder.

[1023] Although inner encoder 130 may be used to provide additional error correction capability, the inventive block decoding techniques described herein may be used for a coding scheme that does not include the inner coding provided by encoder 130. Inner encoder 130 is thus optional, as represented by the dashed box. Also, the data provided to outer encoder 120 may represent data that has been previously coded with a particular code (i.e., instead of “raw” data or information bits).

[1024] The coded data from inner encoder 130 is then provided to a modulator/transmitter (Mod/TMTR) 140, which modulates (e.g., covers and spreads) the coded data to provide modulated data and further conditions (e.g., converts to one or more analog signals, filters, amplifies, upconverts, and quadrature modulates) the modulated data to provide a modulated signal suitable for transmission over a (e.g., wireless) communication channel.

[1025] At receiver unit 150, the transmitted modulated signal is received by an antenna 152 and provided to a receiver/demodulator (RCVR/Demod) 154. Receiver/demodulator 154 conditions (e.g., filters, amplifies, and downconverts) the received signal and digitizes the conditioned signal to provide data samples. Receiver/demodulator 154 may further process (e.g., despread and deconvolve) the data samples to provide demodulated data.

[1026] In the embodiment shown in FIG. 1, the demodulated data is provided to an inner decoder 160 that includes a Viterbi decoder 162 and a de-interleaver 164. Viterbi decoder 162 performs decoding for the convolutional code used at transmitter unit 110, and de-interleaver 164 reorders the decoded bits in a complementary manner to that performed by interleaver 132. The deinterleaved data is then provided to an outer decoder 170.

[1027] Outer decoder 170 includes a CRC checker 172 and a block decoder 174. For each received block corresponding to a coded block transmitted from the transmitter unit, CRC checker 172 checks each row of the received block and provides an indication of whether the row was received correctly or in error (i.e., erased). The CRC-checked block is then provided to block decoder 174, which performs erasure-and-single-error or erasure-only correction block decoding on the block, as described in further detail below. The decoded data from block decoder 174 is then provided to a data sink 176.

[1028] A controller 180 may be used to direct various decoding steps at receiver unit 150. Controller 180 may also be used to implement some or all of outer decoder 170. In this case, the program codes and necessary data may be stored in a memory 182, which is operatively coupled to controller 180.

[1029] FIG. 1 shows a specific embodiment of the transmitter and receiver units capable of implementing various aspects and embodiments of the invention. Other transmitter and receiver designs may also be used and are within the scope of the invention. For example, a receiver unit may be designed to include CRC checker 172 and block decoder 174 but no inner decoder 160.

[1030] FIG. 2 is a diagram that graphically illustrates the various steps for performing erasure-only correction block decoding for a linear block code. An (N, K) linear block encoder codes each "block" of K data symbols (in accordance with a particular set of polynomials if it is a cyclic code) to provide a corresponding codeword of N code symbols. The minimum distance, D, of the code determines the erasure and error correction capability of the block code and the code parameters (N, K) determine the memory requirement. It is known that an (N, K) block code can simultaneously correct for T symbol errors and F erasures in a given codeword, where T and F conform to the following condition:

$$(2T + F) \leq (D - 1). \quad \text{Eq (1)}$$

The inventive block decoding techniques described herein may be used for any code rate. For simplicity, a small code with rate (8, 4) and minimum distance D = 5 is used for the specific example shown in FIG. 2.

[1031] At the transmitter unit, each group of some particular number of data frames is represented by a KxL information block, $i_{K \times L}$. In an embodiment, each row of the

information block corresponds to a respective data frame and includes L symbols for all the bits in that frame. Each symbol (which is represented with a small square box in the information block, $i_{K,L}$) comprises N_B bits, with the specific value for N_B being dependent on the particular block code selected for use.

[1032] The block encoding is performed by pre-multiplying the information block, $i_{K,L}$, with an $N \times K$ generator matrix, $G_{N \times K}$ (step 1). The generator matrix may be derived based on the set of polynomials determined for the selected linear block code. Techniques for determining the polynomials and deriving the generator matrix are known in the art and not described herein. Since each symbol in the information block and the generator matrix may be a multi-bit value, the element-by-element multiplications and additions for the matrix multiplication are performed over the Galois field $GF(2^{N_B})$, where N_B is the number of bits for each symbol. The multiplication of the generator matrix with the information block produces an $N \times L$ coded block, $c_{N,L}$.

[1033] For a systematic (N, K) block code, each codeword includes K data symbols and $(N-K)$ parity symbols that are formed by linear combinations of the K data symbols. Thus, for the example shown in FIG. 2, the coded block, $c_{N,L}$, includes K systematic rows for the K data rows in the information block, $i_{K,L}$, and $(N-K)$ parity rows generated based on the K data rows and the generator matrix. Each of the L columns in the coded block, $c_{N,L}$, corresponds to a respective codeword.

[1034] As shown in FIG. 1, the coded block, $c_{N,L}$, is further processed at the transmitter unit and transmitted to the receiver unit, which then performs the complementary processing to provide a received block, $r_{N,L}$. Each symbol in the received block, $r_{N,L}$, corresponds to a symbol in the coded block, $c_{N,L}$, but may be received in error due to degradation caused by the communication channel.

[1035] At the receiver unit, each row (e.g., each data frame) in the received block may be checked by CRC checker 172 to determine whether or not the row has been received correctly or in error (step 2). If the CRC for a particular row does not check, then that entire row is deemed to have been received incorrectly, even though the error may be to only one symbol in the row, some of the symbols, or all of the symbols. For the example shown in FIG. 2, rows 2 and 4 of the received block, $r_{N,L}$, are marked as

erased since the CRC did not check for these rows. These erased rows are indicated by the rows of shaded square boxes.

[1036] In certain instances, a row may pass the CRC check even though it includes multiple errors. This may occur if the number of errors in the row exceeds the error detection capability of the selected CRC code and the specific values for the errors are such that the CRC (coincidentally) checks. In such a case, the row would be (erroneously) indicated as having been received correctly when in fact it includes multiple undetected symbol errors. An example of such an error row is shown by row 6 in the received block, $r_{N \times L}$, where the three undetected symbol errors are indicated by the three black square boxes.

[1037] A block decoding technique that only corrects for erasures (but not errors) may be performed as follows. First, the rows of the generator matrix, $G_{N \times K}$, corresponding to the erased rows of the received block, $r_{N \times L}$, are also marked as erased (step 2). For the example shown in FIG. 2, rows 2 and 4 of the generator matrix, $G_{N \times K}$, are marked as erased, as indicated by the shaded square boxes in $G_{N \times K}$.

[1038] For the next step in the erasure-only correction block decoding, any K un-erased rows of the received block, $r_{N \times L}$, are selected to form a reduced received block, $r'_{K \times L}$, and the corresponding K rows of the generator matrix, $G_{N \times K}$, are also selected to form a reduced generator matrix, $G'_{K \times K}$ (step 3). For the example shown in FIG. 2, the first four un-erased rows (i.e., rows 1, 3, 5, and 6) of the received block, $r_{N \times L}$, are selected to form the reduced received block, $r'_{K \times L}$, and the corresponding rows 1, 3, 5, and 6 of the generator matrix, $G_{N \times K}$, are selected to form the reduced generator matrix, $G'_{K \times K}$. In this example, one of the selected rows (i.e., row 6) also happens to include undetected symbol errors.

[1039] The reduced generator matrix, $G'_{K \times K}$, is then inverted to derive an inverted (or inverse) generator matrix, $G'^{-1}_{K \times K}$ (step 4). The matrix inversion may be performed in a manner known in the art. The inverted generator matrix, $G'^{-1}_{K \times K}$, is then multiplied with the reduced received block, $r'_{K \times L}$, to derive an initial estimate of the information block, $i'_{K \times L}$ (step 4). An estimate of the information block, $\hat{i}_{K \times L}$, may then be derived

by replacing the erased systematic rows (i.e., rows 2 and 4) of the received block, $r_{N_{dL}}$, with the corresponding rows of the initial estimated information block, $i'_{K_{dL}}$ (step 5).

[1040] If the symbols in all of the rows in the reduced received block, $r'_{K_{dL}}$, are of correct values, then the estimated information block, $\hat{i}_{K_{dL}}$, would be equal to the transmitted information block, $i_{K_{dL}}$ (i.e., $\hat{i}_{K_{dL}} = i_{K_{dL}}$). In this way, the block decoding is able to correct for erasures in the received block, $r_{N_{dL}}$, by using the redundant parity rows (i.e., rows 5 and 6) to recover the erased systematic rows (i.e., rows 2 and 4).

[1041] However, for the example shown in FIG. 2, the undetected symbol errors in the last row of the reduced received block, $r'_{K_{dL}}$, result in errors in the corresponding columns of the initial estimated information block, $i'_{K_{dL}}$. These errors are shown by the black square boxes in columns 4, 7, and 12 of the initial estimated information block, $i'_{K_{dL}}$. Since rows 2 and 4 of the received block, $r_{N_{dL}}$, are erased, these rows are replaced by the corresponding rows 2 and 4 of the initial estimated information block, $i'_{K_{dL}}$. As shown in FIG. 2, each of the replaced erased rows will include as many symbol errors as those in the error row. These errors would then be presented to the next processing stage.

[1042] FIGS. 3A and B graphically illustrate the various steps for performing erasure-and-single-error correction block decoding for a linear block code. For simplicity, the embodiment of FIGS. 3A and B uses the same example used for FIG. 2, whereby the received block, $r_{N_{dL}}$, includes two erased rows 2 and 4, and one error row 6 (which contains multiple undetected symbol errors).

[1043] A block decoding technique that corrects for erasures and a single error may be performed as follows. Initially, each row of the received block, $r_{N_{dL}}$, may be checked by CRC checker 172 to determine whether the row has been received correctly or in error, and the row is marked as erased if the CRC does not check (step 2 in FIG. 2). An un-erased systematic row (e.g., the first such row) of the received block, $r_{N_{dL}}$, is then marked as a pseudo-erasure row (step 3 in FIG. 3A) and is treated like an erased row for some of the subsequent decoding steps (i.e., steps 4 and 5). Next, the rows of the generator matrix, $G_{N_{dK}}$, corresponding to the erased rows of the received block, $r_{N_{dL}}$, are also marked as erased (step 3). For the example shown in FIGS. 3A and 3B,

rows 1, 2 and 4 of the generator matrix, $G_{N \times K}$, are marked as erased, as indicated by the shaded square boxes.

[1044] For the next step in the erasure-and-single-error correction block decoding, any K un-erased rows of the received block, $r_{N \times L}$, are selected to form the reduced received block, $r'_{K \times L}$, and the corresponding K rows of the generator matrix, $G_{N \times K}$, are also selected to form the reduced generator matrix, $G'_{K \times K}$ (step 4). For the exemplary embodiment shown in FIGS. 3A and 3B, the first four un-erased rows (i.e., rows 3, 5, 6, and 7) of the received block, $r_{N \times L}$, are selected to form the reduced received block, $r'_{K \times L}$, and the corresponding rows 3, 5, 6, and 7 of the generator matrix, $G_{N \times K}$, are selected to form the reduced generator matrix, $G'_{K \times K}$. Again for this example, one of the selected rows (i.e., row 6) happens to include undetected symbol errors.

[1045] The reduced generator matrix, $G'_{K \times K}$, is then inverted to derive an inverted generator matrix, $G'^{-1}_{K \times K}$ (step 5). Next, the inverted generator matrix, $G'^{-1}_{K \times K}$, is multiplied with the reduced received block, $r'_{K \times L}$, to derive the initial estimated information block, $i'_{K \times L}$ (step 5).

[1046] For the next step in the block decoding, the pseudo-erasure row (i.e., row 1) is compared against its corresponding row in the initial estimated information block, $i'_{K \times L}$ (step 6). Because of the matrix multiplication, any symbol error in the reduced received block, $r'_{K \times L}$, results in an entire column of symbol errors in the initial estimated information block, $i'_{K \times L}$. Thus, when the pseudo-erasure row is compared symbol-by-symbol against the corresponding row in the initial estimated information block, $i'_{K \times L}$, errors (i.e., unmatched symbols) are detected at columns 4, 7, and 12 since the symbols at these locations do not match.

[1047] A codeword corresponding to a column that contains an unmatched symbol is then selected (step 6). For the exemplary embodiment shown in FIGS. 3A and 3B, the codeword corresponding to the column containing the first unmatched symbol (i.e., column 4) is selected. Computation is then performed to locate the symbol error in the selected codeword. The error location may be determined based on various block decoding techniques known in the art. For example, for a Reed-Solomon code, syndromes are initially computed from the N symbols in the codeword, coefficients of an error location polynomial, $\sigma(x)$, are then calculated from the syndromes, and error

locators may then be computed from these coefficients. For the exemplary embodiment shown in FIGS. 3A and 3B, the symbol error is located at the 6th symbol position in the codeword.

[1048] In the next step, the entire row where the symbol error is located (i.e., row 6) is marked as an erased row, and the pseudo-erasure row (i.e., row 1) is marked as an un-erased row (step 7). The subsequent block decoding may then proceed as described above in FIG. 2. Specifically, any K un-erased rows of the received block, r_{NXL} , (i.e., rows 1, 3, 5, and 7) may be selected to form a new reduced received block, r_{KXL}^* , and the corresponding K rows of the generator matrix, G_{NXL} , are also selected to form a new reduced generator matrix, G_{KXL}^* . As shown in FIG. 3, the row with the (previously undetected) symbol errors (i.e., row 6) is marked as an erased row and is not selected for use.

[1049] The new reduced generator matrix, G_{KXL}^* , is then inverted to derive an inverted generator matrix, G_{KXL}^{*-1} (step 8). The inverted generator matrix, G_{KXL}^{*-1} , is next multiplied with the reduced received block, r_{KXL}^* , to derive a new initial estimate of the information block, i_{KXL}^* . The estimate of the information block, \hat{i}_{KXL} , may then be derived by replacing the erased systematic rows (i.e., rows 2 and 4) of the received block, r_{NXL} , with the corresponding rows of the new initial estimated information block, i_{KXL}^* (step 9).

[1050] Since the symbols in all of the rows of the new reduced received block, r_{KXL}^* , are correct values, the estimated information block, \hat{i}_{KXL} , is equal to the transmitted information block, i_{KXL} (i.e., $\hat{i}_{\text{KXL}} = i_{\text{KXL}}$). In this example, the erasure-and-single-error correction block decoding is able to correct for two erased rows and a single error row in the received block, r_{NXL} , by using the redundant parity rows (i.e., rows 5 and 7) to recover the erased systematic rows (i.e., rows 2 and 4).

[1051] In general, the erasure-and-single-error correction block decoding is achieved by (1) using the redundant parity rows to derive an estimate for an un-erased systematic row, (2) comparing the un-erased systematic row against its estimate to determine the location of any unmatched symbol (or error), (3) locating a symbol error in a codeword (or column) containing an unmatched symbol, (4) marking the entire row

containing the symbol error as being erased, and (5) performing block decoding for the systematic rows based on the un-erased rows of the received block.

[1052] The erasure-and-single-error correction block decoding technique of the invention is less computationally intensive than conventional decoding techniques (partly because error location is performed for only one codeword if an undetected symbol error is subsequently detected with the pseudo-erasure row). The pseudo-erasure row may thus be used to pinpoint the exact codeword (or the exact column in a 2-dimensional block) where an undetected symbol error is located, and the exact location of the symbol error in the codeword (or the exact row of the 2-dimensional block) may be determined by performing block error location on the codeword. The row containing the (previously undetected) symbol error is then marked as erased and erasure-only correction block decoding may be performed on the received block.

[1053] Steps 3 through 6 in FIG. 3A effectively determine the particular column where an undetected symbol error may be found. Other schemes may also be used to determine the error column, and this is within the scope of the invention.

[1054] Moreover, in step 6, the pseudo-erasure does not need to be derived all at once (i.e., for the entire row). Instead, the pseudo-erasure may be derived one symbol at a time and compared to the corresponding symbol in the received pseudo-erasure row. That way, if an error is located in one of the first columns, the remaining pseudo-erasures in the row do not need to be derived.

[1055] The derivation of an initial estimate of the information block, i'_{KdL} , and the formation of an estimated information block, \hat{i}_{KdL} , are conceptual steps. These steps are shown for the example in FIGS. 3A and 3B to provide a better understanding of the block decoding technique. However, in practice, the only rows of i'_{KdL} that are derived are the ones that are erased in the original received block.

[1056] Because only (K+1) good rows (or CRC checked frames) in a received block are needed to perform erasure-and-single-error decoding, it is possible to turn off the front end of the receiver unit (e.g., receiver/demodulator 154, Viterbi decoder 162, deinterleaver 164, and CRC checker 172 in FIG. 1) as soon as (K+1) good rows have been received. This will save power in the receiver unit.

[1057] FIG. 4 is a simplified flow diagram of a process 400 for performing erasure-and-single-error correction block decoding for a linear block code, in accordance with an embodiment of the invention. Initially, a coded block comprised of a number of

codewords is received, and erased rows in the received block are determined, at step 412. The erased rows may be determined by performing CRC check on each row of the received block, or by some other mechanisms.

[1058] A codeword corresponding to a column of the received block containing an undetected symbol error is then identified, at step 414. This undetected symbol error is one that is not included in an erased row of the received block. The column may be identified with the use of a pseudo-erasure row as described above or by some other means. The location of the symbol error in the codeword is then determined (e.g., using any technique known in the art), at step 416. The row of the received block containing the symbol error is then marked as an erased row, at step 418. Block decoding may then be performed for the received block with the newly marked erased row containing the symbol error, at step 420.

[1059] FIGS. 5A and 5B show a detailed flow diagram of a process 500 for performing erasure-and-single-error correction block decoding for an (N, K) linear block code, in accordance with an embodiment of the invention. Initially, a coded block comprised of a number of (L) codewords is received, and erased rows in the received block are determined, at step 512.

[1060] A determination is then made whether the number of erased rows in the received block is greater than $(D-1)$, at step 514. As noted above in equation (1), a properly designed block code may be able to correct for up to $(D-1)$ erased symbols in a codeword. Thus, if the number of erased rows exceeds $(D-1)$, then correction for all the erased rows is not possible since the code distance is exceeded, as indicated at step 516. The process then terminates.

[1061] Otherwise, if the number of erased rows is equal to or less than $(D-1)$, then a determination is made whether the number of erased rows in the received block is greater than $(D-3)$, at step 518. As also noted above in equation (1), a properly designed block code may be able to correct for up to $(D-3)$ erased symbols and one error symbol in a codeword. Thus, if the number of erased rows is equal to $(D-2)$ or $(D-1)$, then erasure-only correction for all the erased rows is possible and may be performed starting at step 522.

[1062] Otherwise, if the number of erased rows is equal to or less than $(D-3)$, then a determination is made whether the number of erased systematic rows in the received block is greater than $(K-1)$, at step 520. If one of the systematic rows is to be used as a

pseudo-erasure row to locate the column of an undetected symbol error and if no un-erased systematic row is available for use as the pseudo-erasure row, then erasure-only correction for all the erased rows is still possible and may again be performed starting at step 522.

[1063] Otherwise, if the number of erased rows is equal to or less than (D-3) and at least one un-erased systematic row is available, then erasure-and-single-error correction block decoding may be performed on the received block, starting at step 532. In general, erasure-and-single-error correction block decoding may be performed whenever possible, and erasure-only correction block decoding may be performed if possible and if erasure-and-single-error correction block decoding cannot be performed.

[1064] To perform erasure-only correction block decoding, a reduced generator matrix, $G'_{K \times K}$, is initially formed by selecting K rows of the generator matrix, $G_{N \times K}$, corresponding to any K un-erased rows of the received block, $r_{N \times L}$, at step 522. The reduced generator matrix is then inverted, at step 524. The erased systematic rows of the received block are then derived from the corresponding rows of an initial estimate of the information block, $i'_{K \times L}$, which is derived by multiplying the inverse generator matrix, $G'^{-1}_{K \times K}$, with a reduced received block, $r'_{K \times L}$, formed by K corresponding un-erased rows of the received block, $r_{N \times L}$, at step 526. If there are no undetected symbol errors in the rows in the reduced received block, $r'_{K \times L}$, then the initial estimated information block, $i'_{K \times L}$, would include corrections for all the erased systematic rows in the received block. The erasure-only correction block decoding then terminates.

[1065] To perform erasure-and-single-error correction block decoding, the row location of an undetected symbol error in the received block is initially determined. This may be achieved as follows. First, an un-erased systematic row (e.g., the first such row in the received block) is selected as a pseudo-erasure row, at step 532. This row is treated as an erased row in the following steps to determine the undetected symbol error location. A reduced generator matrix, $G'_{K \times K}$, is then formed by selecting K rows of the generator matrix, $G_{N \times K}$, corresponding to any K un-erased rows of the received block, $r_{N \times L}$, at step 534. The reduced generator matrix is then inverted, at step 536. An estimate of the pseudo-erasure row is then derived by multiplying the inverse generator matrix, $G'^{-1}_{K \times K}$, with a reduced received block, $r'_{K \times L}$, formed by K corresponding un-

erased rows of the received block, r_{NKL} , at step 538. Since only the estimate of the pseudo-erasure row is needed, the matrix multiplication of G'_{KK}^{-1} with r'_{KdL} may be performed for only one row of G'_{KK}^{-1} (and not the entire matrix) against all columns of r'_{KdL} .

[1066] The pseudo-erasure row is then compared against its estimate, at step 540. The comparison may be performed symbol-by-symbol until a first unmatched symbol is detected. A determination is then made whether or not there are any unmatched symbols in the pseudo-erasure row, at step 542. If there are no unmatched symbols, then there are no undetected symbol errors in the received block (or there is more than one symbol error in a single column), and the process proceeds to step 526 to derive the erased systematic rows.

[1067] Otherwise, if there is at least one unmatched symbol in the pseudo-erasure row, as determined in step 542, then there are undetected symbol errors present in the received block and the row location of these errors is next determined, at step 544. This may be achieved by identifying the location of an unmatched symbol (e.g., the first unmatched symbol) in the pseudo-erasure row, retrieving the codeword (or column of the received block) containing this unmatched symbol, and performing error location on this codeword based on any technique known in the art. The symbol error in the codeword is thus located, and the entire row of the received block containing this symbol error is marked as an erased row, at step 546.

[1068] A determination is then made whether or not the pseudo-erasure row is the one with the symbol error, at step 548. If the answer is yes, the error row was already excluded as the pseudo-erasure when the reduced received block and reduced generator matrix were formed. Therefore G'_{KK}^{-1} is correct and r'_{KdL} does not contain any undetected errors. The process then proceeds to step 526 where the erased systematic rows are derived. Otherwise, a new reduced generator matrix, G''_{KK} , is then formed by selecting K rows of the generator matrix, G_{NK} , corresponding to any K rows of the received block, r_{NKL} , that are still un-erased, at step 550. The new reduced generator matrix is then inverted, at step 552.

[1069] The process then proceeds to step 526 where the erased systematic rows of the received block are derived from the corresponding rows of the new initial estimate of the information block, i''_{KdL} , which is derived by multiplying the new inverse

generator matrix, $G_{K \times K}^{*-1}$, with a new reduced received block, $r_{K \times L}''$, formed by K corresponding un-erased rows of the received block, $r_{N \times L}$.

[1070] Various modifications to the process shown in FIG. 5 may be made. For example, it may be possible to iterate steps 532 through 546 multiple times to detect for multiple error rows.

[1071] The block decoding techniques of the invention may be advantageously used for a concatenated code comprised of an error detection code and a block code. The error detection coding may be performed in one (e.g., horizontal) dimension of an information block and the block coding may be performed in the other (e.g., vertical) dimension of the information block. The inventive block decoding techniques may be used to identify and locate errors that the error detection decoding fails to catch, and to remove these symbol errors from the block decoding process. Thus, the inventive block decoding techniques can ameliorate problems caused by failure to detect errors, which cannot be handled by conventional block decoding techniques that can only correct for erasures but not errors.

[1072] The inventive block decoding described herein is computationally efficient and may be well suited for a software-based implementation executed on a microprocessor. To quantify the computational efficiency, the number of multiply-accumulate operations performed by some of the steps in the flow diagram in FIGS. 5A and 5B is provided below:

<u>Steps</u>	<u>Number of Multiply-Accumulate Operations</u>
524, 536, 552	KP_{sys}^2 (for each step)
538	KL
544	$(P+2)(N-1) + P(P+1) + 1$ (for a Reed-Solomon code)
526	KLP_{sys}

where

N is the codeword length;

K is the number of information symbols in a codeword;

R is the number of parity symbols (i.e., $R = N - K$);

L is the number of codewords per coded block;

P is the number of erased rows in the received block; and

P_{sys} is the number of erased systematic rows in the received block.

[1073] The inventive block decoding techniques described herein may be used for various communication and data transmission systems. For example, these techniques may be advantageously used for wireless communication systems such as CDMA, FDMA, and TDMA systems. Moreover, the decoding techniques described herein may be implemented in a terminal (e.g., a wireless device such as a cellular telephone) for the downlink (i.e., the forward link) or at a base station or access point for the uplink (i.e., the reverse link) in a wireless communication system.

[1074] As noted above, the inventive block decoding techniques described herein may be implemented by various means. For example, these techniques may be implemented in hardware, software, or a combination thereof. For a hardware implementation, the elements used to implement some or all of the block decoding may be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-controllers, microprocessors, other electronic units designed to perform the functions described herein, or a combination thereof. The hardware (e.g., an ASIC or DSP) may include one or more functional units that collectively perform the processing described above to implement the inventive block decoding. For example, one unit may be provided to perform the row-wise error detection (e.g., CRC checker 172), another unit may be provided to perform the column-wise block decoding (e.g., block decoder 174), and so on.

[1075] For a software implementation, the inventive block decoding techniques may be implemented with modules (e.g., procedures, functions, and so on) that perform the functions described herein. The software codes may be stored in a memory unit (e.g., memory 182 in FIG. 1) and executed by a processor (e.g., controller 180). The memory unit may be implemented within the processor or external to the processor, in which case it can be communicatively coupled to the processor via various means as is known in the art.

[1076] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without

departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.